U.S. UTILITY PATENT APPLICATION

IN THE NAME OF

Nancy J. Rabenold
James A. Simmons
Paul D. Sper

Filed: May 25, 2001

REMOTE BIDDING SUPPLEMENT FOR TRADITIONAL LIVE AUCTIONS

Claiming priority from U.S. Provisional Patent Application No. 60/207,030 Filed May 25, 2000

Attorney Docket No. 16628-14

REMOTE BIDDING SUPPLEMENT FOR TRADITIONAL LIVE AUCTIONS

This application claims the benefit of co-pending provisional application serial number 60/207,030, filed May 25, 2000. This application also incorporates by reference the computer program listing appendix ("System Software") Copy 1 and Copy 2 containing the files having the dates of creation and size in bytes set forth on pages 70 to 73 hereof.

The remote bidding supplement for traditional live auctions of the present invention is a software-based product that provides the capability for a user to instantaneously interact with and enjoy the emotion and enthusiasm of a traditional, live auction (view items for sale, view live bidding, hear the auctioneer calling bids, view the activities of the onsite participants, make bids, buy items) from a position that is physically remote from the live auction.

BACKGROUND OF THE INVENTION

Traditional-style auctions are ignoring a significant market – the physically remote purchasers who will purchase an item without being physically present to kick the tires, feel the smoothness of a vase, hear the roar of a diesel engine or authenticating an ancient item.

Currently, there are two types of remote auction systems. The first type of remote auction system has no "live" auctioneer and the entire bidding audience must be connected to the network or system. In this case, the network computer

or server acts as the auctioneer, accepting bid values from the connected audience with associated time stamps based upon bid receipt by the server. Each bid is either accepted or rejected by the server and the bidder (sometimes the entire audience) is notified of its acceptance or rejection. All of the items for sale in this type of auction are generally available for the entire duration of the auction and each item has a specified end time after which no bids will be accepted.

The second type of remote auction system is much like the first; however, it may or may not have a "live" auctioneer. The main difference from the first type of remote auction system is that each item for sale is not available at the same time; rather, the auction moves from item to item and depending upon the bidding activity and upon either the server's or "live" auctioneer's choice, the item is sold and the event moves on to the next item on the list.

SUMMARY OF THE INVENTION

The present invention allows for prospective auction bidders to participate both in person as well as in a remote capacity. The present invention enables an existing traditional-style auction company to utilize technology that allows an auction to be conducted in the traditional style, generating the emotion and enthusiasm in the local audience, leaving the auctioneer in total control of the sale, while providing the opportunity for other bidders to "attend" the auction event remotely (e.g., via the Internet), sharing the same emotion and enthusiasm

as the local audience and participating in the bidding process without disadvantage, just as if those physically remote bidders were sitting in the local auction audience.

The present invention provides the catalyst for changing the live auction process by ensuring an environment in which all parties – whether in person or in a remote location–can participate as one audience without impact to the natural flow, speed and excitement of the live auction. Only with the present invention can remote bidders compete – without disadvantage – against live floor bidders in an instantaneous bid environment while realizing the true emotion and enthusiasm of the traditional auction setting. The remote bidding system of the present invention was specifically designed to enhance the live auction process rather than to alter it. The control of the auction always remains with the live auctioneer and the auction company. The auctioneer can effectively and efficiently take bids from either the local audience or from the remote audience (e.g., via the Internet). As in all traditional live auctions, the auctioneer can accept or reject any given bid from the local or remote audience. The system in effect greatly enlarges the potential pool of bidders by eliminating time, geographic or travel constraints on behalf of the bidder and quickly increases the potential reach of the auction company from a regional business to potentially a nationwide or worldwide business.

Crucial to successful seamless integration of a remote auction audience with the local or onsite audience are features of the present invention that allow the remote bidder to rapidly make a purchase decision – not alienate the bidders who took the time to actually come to the auction event – while instilling the confidence of all parties (onsite, remote and the auction company) in the integrity of the process. The remote bidding system of the present invention accomplishes this through the following systems of the present invention.

1) Audio/Video System – The actual emotion and enthusiasm of a traditional-style auction event is transferred to the remote bidder or participant through streaming audio and video technology. The audio is transmitted from the auction site to the remote participants in one (1) second or less through the network and the video is transmitted in real-time at a frame rate that supports a 56K modem connection to the network. Competing streaming live audio and video technologies of today utilize a buffering method at the encoding and/or the receiving end to achieve an acceptable level of quality for audio and video. In a traditional-style auction environment (i.e., dealer-only automobile auctions), an item may be sold every 20 to 30 seconds with 20 to 30 bids. Buffering at the encoding and/or the receiving end typically adds 7 or more seconds in delay to the audio and video that would place the remote participant at an extreme disadvantage. The present invention removes the buffering without sacrificing quality and with a resulting delay of only one (1) second or less.

2) Bid/Clerk Systems –A Bid System controls the instantaneous interactions between the remote bidders, a Clerk System, and a Marquee System. The Clerk System controls the sequencing of items to be sold through the auction and controls the auction bidding process, both live and remote, for each item to be sold. The Marquee System displays instantaneously auction bid information for each item being sold at auction.

a. Cherokee Bid Engine – The Cherokee bid processing algorithm within the Bid System allows the auction to proceed at a very fast pace (in excess of 140 items per hour with sometimes as many as 30 bids per item). This algorithm uses a fixed increment predictive algorithm to present bid choices to both the auction clerk as well as the remote bidders. The Cherokee model also assigns the default high-priority to the remote bidders, but allows the auctioneer and clerk to change for any specific bid.

b. Iroquois Bid Engine – The alternative Iroquois bid processing algorithm within the Bid System allows the auction to proceed at a fast pace while adding flexibility in its fixed increment predictive algorithm to accommodate a range of fixed increments depending on the actual last high bid. The Iroquois model assigns the default high-priority to the onsite/local bidders, while allowing the auctioneer and clerk to change any specific bid.

c. Apache Bid Engine – The alternative Apache bid processing algorithm within the Bid System addresses many of the auction segments

that do not operate with a fixed increment policy. In this model, the clerk simply follows the auctioneer's "asking price" chant and allows remote bidders to submit a bid for that price. The Apache model allows for flexibility in choosing the default high-priority bid to be either the onsite/local bidder or the remote bidder.

3) Marquee System – Another critical aspect of the present invention is the design of the Marquee that is physically placed at the auction site. The Marquee is used to:

a. Identify incoming remote bids to the auctioneer

b. Identify incoming remote bids to the onsite/local audience to create confidence that there is a remote bidder and the identity of the remote bidder; and

c. Identify to the onsite/local audience the item being sold as well as the current high bid amount accepted (the onsite/local audience starts to bid off the Marquee).

4) Data Mining – The data mining bid log processing capability of the present invention is a unique function that provides auction companies with the ability to quickly analyze the auction event's activity to assess national and international market value of the sale items, the participation of any or all remote bidders or the incremental value-add brought by the remote auction capability. When the data mining bid log is evaluated in conjunction with the pre-sale catalog and the condition report of the item, an instantaneous assessment can be made of how

the features and the characteristics (e.g. damage to an automobile) of the sale item may impact the re-marketability of the item.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic of the primary elements of the remote bidding supplement of the present invention.

Figure 2 is a schematic of the dataflow within the remote bidding supplement of Figure 1.

Figure 3 is a schematic of the dataflow within the Audio/Video Capture System of the present invention.

Figure 4 is a schematic of the Audio/Video Capture System of the present invention.

Figure 5 is a schematic of the Audio/Video System of the present invention.

Figure 6 is an illustration of a Marquee Display from the Cherokee Bid Engine of the present invention.

Figure 6A is a schematic of the dataflow for the Marquee System of the present invention.

Figure 7 is a schematic of the Marquee System process flow of the present invention.

Figures 7A/7B illustrate initial Marquee System logon displays, Figure 7A being generated by the Bid System of the present invention, and 7B being the display of 7A after "user name" and "password" have been entered.

Figure 7C illustrates a Marquee System display with a first DISMISS message.

Figure 7D illustrates a Marquee System display with a second DISMISS message.

Figure 7E illustrates a Marquee System display after logon is complete.

Figure 8A illustrates a Bidders Display from the Cherokee Bid Engine of the present invention.

Figure 8B is a schematic of the dataflow for the Bidder System/Device of the present invention.

Figure 9 is a schematic of the Bidder System process flow of the present invention.

Figure 9A illustrates a Bidder Display after a URL connection is made.

Figure 9B illustrates a Bidder Display with "user name" and "password" entered.

Figure 9C illustrates a Bidder Display with a DISMISS message box.

Figure 9D illustrates a Bidder Display after the login sequence is completed.

Figure 10A illustrates a Clerk Display.

Figure 10B is a schematic of the dataflow for the Clerk System of the present invention.

Figure 10C is a schematic of the Clerk System process flow of the present invention.

Figure 11A/B illustrates the initial Clerk System logon displays.

Figure 11C illustrates the Clerk System Display with a first DISMISS message box.

Figure 11D illustrates a Clerk System Display with a second DISMISS message box.

Figure 11E illustrates a Clerk System Display after the login process is complete.

Figure 12 is a schematic of the Bidding Process activation for an item to be sold at auction.

Figures 13A/13B/13C illustrate examples of the Marquee, Clerk and Bidder Displays following the initiation of the first item in a sequence to be sold at auction.

Figure 14 is a schematic of the entry of a starting bid value.

Figures 13D/13E/13F illustrate the Marquee, Clerk and Bidder Displays after entry of a starting bid.

Figure 15 is a schematic of the process for entry of a floor bid.

Figure 16 is a schematic of the process for entry of a remote bid.

Figures 13G/13H illustrate the examples of the Marquee and Clerk Displays when there is a pending remote bid.

Figure 17 is a schematic of the process for acceptance of a remote bid.

Figure 13I illustrates an example of a Bidder Screen when there is an accepted remote bid.

Figure 18 is a schematic of the process to override a remote bid.

Figure 19 is a schematic of the process for a sold bid.

Figures 13J/13K illustrate the Bidder and Clerk Display screens after SOLD.

Figure 20 is a schematic of the process for a remote user to request purchase information from the Cherokee Bid Engine.

Figure 21 is a schematic of the process for the clerk to delete a bid.

Figure 22 is a schematic of the process for the clerk to send a message.

Figure 22A illustrates an example of a Message Screen.

Figure 23A/23B illustrates examples of Bidder Screens and Figure 23C is a Clerk Screen from the Mohawk Bid Engine of the present invention.

Figures 24A/24B illustrate examples of a Bidder Screen and a Clerk Screen from the Iroquois Bid Engine of the present invention.

Figures 25A/25B/25C illustrate examples of Clerk, Marquee, and Bidder Displays from the Apache Bid Engine of the present invention.

Figure 26 illustrates an example of a Clerk Screen from the Apache Bid Engine.

Figure 27 illustrates an example of an updated Bidder Screen from the Apache Bid Engine.

Figure 28 is a schematic of the Audio/Video System of the present invention.

Figure 29 is a schematic of the process flow of the Audio/Video System of the present invention.

DETAILED DESCRIPTIONS OF THE PREFERRED EMBODIMENTS

1 SYSTEM OVERVIEW

The uniqueness of the remote bidding supplement of the present invention is its ability to perform the remote bidder interactions in an instantaneous environment independent of the distance between the remote bidder and the live auction.

The remote bidding supplement of the present invention requires specific hardware and software products to perform these functions:

Three systems, co-located, to perform the control functions of the auction:

A/V System to receive the audio/video stream from an A/V Capture System at the auction and retransmit this stream instantaneously to each of the remote bidders attending the auction. This is part of the System Software.

Bid System to control the interactions between the remote bidders, Clerk System, and a Marquee System. This is part of the System Software.

Catalog System to maintain the pre-sales data on items to be sold (this function may be performed by either the A/V System or the Bid System referenced above). In the normal auction configuration the pre-sales catalog information is kept on the Catalog System.

A Marquee System to display current bid information, from either floor or remote bidders, to the gallery/auctioneer/ringmen at the live auction.

A Clerk System that controls the sequencing of items through the auction and controls the auction bidding process for each item to be sold.

An A/V Capture System to provide the audio/video stream from the camera and sound system at the auction to the A/V System controlling the transmission of the stream to the remote bidders. Specific audio and video capture cards are required for this function. This is part of the System Software.

The remote bidding supplement of the present invention performs the audio/video streaming and the remote bidder/clerk interaction as two independent functions:

The A/V Capture System utilizing the required hardware cards installed in a computer system encapsulates the audio/video stream. This data is transmitted to a specific A/V System where it is re-encapsulated and broadcast to each of the remote bidders logged onto the system. This function is performed independent of the Marquee/Clerk/Bid Systems.

The auction bidding process is controlled by the Bid System and the Clerk System. Data for each item to be sold is extracted from the system maintaining the pre-sales information prior to the auction start, a subset is

created on the Bid System, and is broadcast to all remote bidders and the Marquee System as each item is auctioned. A starting bid is established on the Clerk System and then bids are accepted from floor or remote bidders. Status is transmitted to the Marquee System and all bidders logged onto the auction, and logs are maintained identifying all activity performed by the clerk including status of each bid made by a remote bidder.

A "bid engine" algorithm on the Bid System controls the remote bidding process. There are four main bid engines that can be utilized. The primary functions of these engines are identical. Differences are in the areas of automatic versus clerk controlled acceptance of a remote bid, bid increments used by the Clerk and Bidder Systems, ability to enter starting bids from a remote bidder, and display formats. These engines are identified as:

CHEROKEE

IROQUOIS

MOHAWK

APACHE

The system configuration (Figure 1) identifies the primary relationships between the elements of the remote bidding supplement of the present invention. The three primary systems work together to support the remote bidding supplement of the present invention. Each remote bidding supplement of the present

invention requires an A/V System, Clerk System, and Marquee System that are assigned to an "area" within the Bid System called an environment. Bidders logging onto an auction are assigned to that same environment. The data flow associated with the system of Figure 1 is shown in Figure 2.

1.1 SOFTWARE GENERIC STRUCTURE

The following descriptions summarize the development structure utilized for the elements of the remote bidding supplement software required for the systems supporting the auction process.

1.1.1 COMPONENT DESCRIPTION

The code base for the Marquee, Clerk, and Bid Systems interface is mostly shared. Controls have been made for each main aspect of the user interface (buttons, labels, text boxes, etc). These controls deviate from the standard JAVA AWT controls because their look and functionality needed to be integrated with the rest of the AMS software presentation. The controls appearance on the screen is highly configurable in the html pages that load the applet. If one looks at the html (ringman-admin.htm, ringman-bidder.html and ringman-quack.html) more detail will be self evident. Classes were implemented per needed feature. The classes listen for events on the controls and respond appropriately by sending information to the network module, changing the state of other classes, or changing the state of other controls.

1.1.2 DATABASE

The database is currently implemented in POSTGRES.SQL. Tables and are used for logging the bidder events (proposed, accepted, rejected bids), tracking the sale inventory (item descriptions, etc), and tracking user accounts (username, password, etc). For details see the POSTGRES initialization scripts (files ending in **.psql** which initialize the database for the Bid System).

1.1.3 BID SYSTEM/NETWORK

Current networking is implemented in UDP, original versions were implemented in TCP, which was found to be too latent (3 second latencies in cases of certain packet transmission failures) in testing to date. The current UDP system re-implements most of TCP to get around this. Details of the implementation can be found in the source file "RUDP C" (which stands for reliable UDP). The main problem with this solution to date has been the complexity required in both the Bid Device (client) and Bid System. More simple networking code may replace the RUDP implementation in future upgrades.

Parts of the Bid System use CGI (common gateway interface) programs for relaying information from an SQL database to the user. The Bid System structure is to accept queries, then look up the appropriate information from the SQL database or its internal cache, and then to determine a response and send it to a

group of bidders it has registered as being interested in that type of message. In this process it may alter the state of the database.

1.1.4 DEVELOPMENT/ OPERATING ENVIRONMENT

The Bid System was developed and operates on Redhat Linux 2.0. The project is compiled in the standard way with GNU CC for C portions and JAVAC for JAVA portions. The PSQL C library must be linked with the final Bid System executables. The Bid System should be portable to other Unix (POSIX) environments for development, testing, and operation; this may require modification, however.

1.1.5 PLATFORM

The Bid System currently runs on Redhat Linux 2.0. It was written to be portable within POSIX and BSD sockets. It will probably require modification before it will run under other environments though. Such modification should be straight forward and trivial, but to date no porting efforts outside of Linux have been made [no compatibility with non-POSIX conformant systems or systems that do not implement BSD sockets is either expressed or implied]. The Bid System uses up resources proportional to the amount of users and size of the database it is serving.

The Bidder Device (client) software should be portable to any machine that can run a JVM. However, it has only been tested to date under Windows/Netscape

on Pentium II/400mhz/192MB RAM and better machines. Limited testing on machines lesser than this has revealed performance deficits possibly in the JVM implementation or some aspect of the design. Testing on other JVMS has indicated JVM portability issues which look to be resolvable but will require modification to the structure of the client applet.

1.1.6 LANGUAGES

The ringman server is implemented in C. The code was written to conform to (or at least be portable to any implementation of) the POSIX standard, except for networking sections which conform with BSD sockets. The code was written and tested in the Redhat Linux 2.0 environment with the GNU C Compiler 2.7.2 and has not been tested to date in any other environments. Compatibility with other environments including, but not limited, to previous and future versions of Linux, Windows, or any commercial UNIX system is neither expressed nor implied. UNIX/C was chosen because it is the standard environment for Internet applications.

The Online Ringman client is implemented in Java. Java was chosen because it is currently the only widly used internet WORA (write once run anywhere) package. Current versions of the Ringman System have only been tested under the JVM distributed with Netscape Navigator.

Future plans include moving the client application to Windows/C, which limits the user base to windows users but within that user base should perform faster and more reliably and remove dependence on JVM implementations.

Portions of code that access the SQL database were written with the C version of the postgres SQL library. CGI programs run under the Apache web server. CGI was chosen because it was the simplest way to get information to a client. Since the program is written in Java, it was considered likely that the user would have access to a web browser with CGI capabilities: utilizing these capabilities was an easier, simpler, more extendable interface than adding portions to the Java code base ad hoc.

2 AUDIO/VIDEO STREAMING

The Audio/Video (A/V) Capture System (Figure 3 and Table 1) receives data from an audio device via a sound card in the A/V Capture System and the video signal/device via an Osprey video card in the A/V Capture System. The data is blocked for transmission to the A/V System. The A/V System then integrates the data into an audio/video stream that is broadcast to a communications port on the A/V System. Bidders that have logged onto the system automatically receive the data from this port, which is then displayed on the bidders device and broadcast through the speakers (if available) on the bidder's device.

This process is detailed in Figure 4 and Figure 5 and Section 3.1 (Software Description).

Table 1

MINIMUM A/V CAPTURE SYSTEM CONFIGURATION

IBM Compatible - Pentium III 450 MHz
128 MB SDRAM PC100 (256 MB preferred)
CD ROM
6.4 GB Hard disk drive
1.44 MB Floppy drive
3COM 905 BTX NM 10/100 NIC
Standard keyboard and mouse
VGA Display controller (1024 x 768)
Osprey 100 Video capture card
Sound-Blaster PCI 128 sound card
15" Monitor
Uninterruptible Power Source Converter

Custom configured Red Hat LINUX 6.x Operating System
AMS streamer control software

Video Signal
Audio Signal

LAN connection to a high bandwidth connection to A/V System

The A/V Capture System is used to (a) convert the signal from the video source to a digital stream, which is then transmitted to the A/V System on a continuous basis, and (b) convert the signal from the sound source to a digital stream, which is then transmitted to the A/V System on a continuous basis.

The unit requires a specific video capture card and a specific audio capture card in the A/V Capture system to perform these two functions. The base operating system utilized by this unit is LINUX.

2.1 SOFTWARE DESCRIPTION

The software description for the Audio/Video Streamer is contained in the Audio/Video System Overview, infra.

3 DISPLAY UPDATE PROCESS

The Marquee, Clerk, and Bid Systems displays utilize a basic website technology as the basis for displaying information specific to their functions. Each system connects to a unique URL as part of the login process for the system. Once the login process is complete, the system is linked to that website address. At that point, a base display is generated at specific CRT coordinates. The base display contains the background frame with dynamic display areas blank. This is done to reduce the size of the data packets required when an operator/system action is taken during the normal system operation. From this point forward, small data packets are sent to specific cursor addresses on the appropriate display. For example, when a floor bid is entered on the Clerk System

The Marquee display is updated with the specific bid amount and bidder #

The Clerk system is updated with new bid increments plus a log message

The Bid System is updated with new bid buttons plus last bid value

The second factor that is related to reducing the update time required for display changes is the ability to "broadcast" the same data to all systems connected to a particular URL, primarily the bidder devices. Each system performing a particular function receives the same data at the same time from the Bid Server perspective. The only delay in receipt of the information by the Bid System is the inherent delay in the distance and method of the transmission over their respective communication links.

Throughout the bid processing, movement of the screen cursor to a "selectable bar/button/etc." causes the selected item to either change color (e.g., yellow to red) or change from its idle color to no color (i.e., the frame color).

The displays generated by the system can be changed either at compile time or within the HTML code used to output the static portions of the display. For example, NEXT ITEM can be replaced with NEXT VEHICLE for a Clerk display associated with an automotive auction; REMOTE BID could be replaced with INTERNET for a specific internet auction; the $ in activity log messages and on bid buttons can be changed to £ for an auction in the United Kingdom.

4 BID SYSTEM AUCTION PREPARATION

Prior to the start of each auction, it is necessary to upload catalog data and user data to the Bid System specific to the scheduled auction. A subset of the catalog data is utilized to generate the data that is displayed on the lower left portion of the bidder display during the auction. The user file is used to identify the access

each "bidder" has to the bid process (spectator, bidder with credit limit, clerk, marquee).

4.1 CATALOG PROCESSING

The pre-sales catalog data is maintained as a separate data set for each auction. The maintenance of this data is not part of the process of the present invention. However, specific action must be taken prior to the sale to execute a script that retrieves the data file from the Bid System or the Catalog System and creates a "ringman sub-set" on the Bid System (**./update_db.sh**). This subset is then used by the bid engine to display the lower left data for each item during the bidding process.

Once the data is created it is necessary to execute three scripts on the Bid System for this environment:

resetbidlogs – sets the bidders log and bid log file pointers to the beginning of file

resetnohup – system command for LINUX

restartserver – sets the bid engine to start at the first item in the database (a sequence number within the database establishes the order in which items are processed by the bid engine)

4.2 UPDATE_DB.SH SCRIPT PROCESS

The **update_db.sh** script creates the ringman subset on the Bid System. The ringman subset contains the "lower left" data for the bidders screen and the data

for the Marquee related to each item to be auctioned. The resulting table structure is in run number order.

The script extracts the required inventory data from the inventory file provided to the Catalog System and the required condition report data from the damage file provided to the Catalog System. The inventory data and damage data are subsets of the total data provided for each item. The data extracted is dependent on the format defined by the auction and subsequently must be "specifically coded" for that auction.

4.3 RESET BID LOGS SCRIPT PROCESS

The **resetbidlogs** script resets the bidders log and bid log pointers to beginning of file and clears data currently in the file. This ensures a null file for the start of each auction. If the script is not executed (operator controlled), prior data is not cleared and is still part of the current file.

4.4 RESTARTSERVER SCRIPT PROCESS

The **restartserver** script resets the internal pointers for the auction selected (via logon/password) to the start of the ringman subset such that the system automatically starts at run number 1 when the Clerk System is activated.

4.5 USER FILE UPLOAD

The **user file** contains access information for each user scheduled to connect to the Bid System during an individual auction. Table 2 identifies the format of the

user file maintained separate from the bid server as a TAB delimited file. The data fields in the file are:

Field1 = user ID, this is the "logon" user name

Field2 = password

Field3 = user type; 1 = bidder, 2 = clerk, 4 = marquee

Field4 = credit limit for this auction

Field5 = field set to $0; during the auction, this field is updated on the Bid System to contain the total value of items purchased by each bidder

The file is first uploaded to the specific environment assigned to this auction without a file extension. The auction administrator then executes the **updateusers** script, which creates the internal tables the bid engine utilizes during the live auction.

Table 2 – User File Format

5011748	84711053	1	900000	0		user with $900,000 credit limit
656	101265	1	999999999	0		user with unlimited credit limit
52965	3573179	1	20000	0		
5033281	182305	1	500000	0		
5050186	6810505	1	200000	0		
5032832	2382305	1	50000	0		
5032025	5202305	1	500000	0		
5050658	8560505	1	25000	0		
5025609	9065205	1	100000	0		
5042804	4082405	1	20000	0		
5004057	7504005	1	200000	0		
5032640	99983126	1	250000	0		
5045095	5905405	1	250000	0		

5011587	7851105	1	30000	0		
5029884	89583277	1	30000	0		
10006	6006	1	0	0		user with $0 credit limit = spectator
10015	150015	1	0	0		
63046	64036	1	999999999	0		
5061648	8461605	1	999999999	0		
63056	65036	1	999999999	0		
5035425	55245305	1	75000	0		
5057863	3687505	1	999999999	0		
5058742	2478505	1	999999999	0		
5059001	9500051	1	999999999	0		
clerk	servnet	2	0	0		clerk accesss
marquee	servnet	4	0	0		marquee access
user1	iytdirdir	1	0	0		
user2	abcdef	1	0	0		

4.6 UPDATEUSERS SCRIPT PROCESS FLOW

The **updateusers** script is executed on the Bid System to import data from an external file into the internal tables used to verify logon/password combinations and determine the type of access a user has. The input file must be tab delimited and contain the data referenced in Table 2. The table constructed in the Bid System contains the same fields with no added data.

5 MARQUEE SYSTEM

The Marquee System is the visual link between the auctioneer/ringmen, live gallery, and the remote bidder. The system employs audio and visual prompts to the auctioneer to indicate that a bid has been made by the remote bidder. This system is a display only function that is controlled by the Bid System based on

inputs from the bidder (entry of a bid), the Bid System (automatic acceptance of a bid), or the Clerk System (operator acceptance of a bid). The control sequence is then reset once the current bid has been processed. The Marquee System is linked directly to an output port of the Bid System such that messages (data packets) from the Bid System are automatically broadcast to the Marquee System for output to the display device.

The display indicates the current run # (item) being sold and the current bidder ID (ID from the user file for an remote bidder or "floor bidder" for any bidder at the auction) and amount (whether the bidder is remote or at the auction). When a remote bid is received, the Marquee System flashes the display and beeps to indicate a remote bid has been made plus identifies the remote bidder ID and the bid amount. Based on the type of bid engine installed, this remote bid is either (a) accepted automatically or (b) requires an overt action by the Clerk System to be accepted.

Figure 6 illustrates an example of a Marquee Display for the Cherokee Bid Engine.

Figure 6A identifies the configuration requirements for the Marquee System (see also Table 3) and Figure 7 depicts the process flow for the system.

Table 3

MINIMUM MARQUEE SYSTEM CONFIGURATION

IBM Compatible - Pentium 200MHz

32 MB RAM

CD ROM

4.0 GB Hard disk drive

1.44 MB Floppy drive

3COM 10/100 Ethernet NIC

Standard keyboard and mouse

VGA Display controller (1024 x 768)

Sound-Blaster compatible sound card with speakers*

(*Compaq sound card is not compatible)

RGB (SVGA) to NTSC converter (if display accepts NTSC)

Uninterruptible Power Source/Converter

Windows 95, 98, or NT operating system

Netscape 4.05 or later

Display Device

Figures 7A/7B illustrate initial Marquee System logon displays; the first generated by the Bid System while the second shows the display after "user name" and "password" have been entered. Figure 7C illustrates a Marquee System display with first DISMISS message; the operator clicks on DISMISS to eliminate message box. Figure 7D illustrates a Marquee System display with second DISMISS message; the operator clicks on DISMISS to eliminate message box. Figure 7E illustrates a Marquee System display after login is complete, with no action taken by clerk at this point.

6 BIDDER SYSTEM

The Bid System is used by a remote bidder to view the events at the live auction via the audio/video stream transmitted to each logged on bidder. In addition, the bidder screen displays a sequential history of the bids as controlled by the Clerk System. During the bidding for each item, the Bidder Device allows the user to enter a bid for that item which is transmitted to the Bid System controlling the auction. Recognition of the bid and whether or not it is accepted by the system/clerk is returned to the bidder and the Bid System is then ready to accept another bid from that remote user. The acceptance of the bid and the display returned to the user is controlled by the Bid System and the Clerk System in different combinations based on the install option selected by an auction. The Marquee System is independent of these actions and only receives "website updates" based on their actions. The specific interactions between the Clerk and Bid Systems are defined in the Bid Engine Processes section infra.

The display used by the bidder is reset at three basic points in the process:

1. The base display is generated and made available to all bidders as each logs onto the required URL. At this point, the bidder display contains the audio/video stream from the A/V System plus the base frame for the bid buttons.
2. When the Clerk enters NEXT ITEM, the display is updated to contain the information for the item in a pre-set area of the display. This data area is not updated until the next item is selected for the bid process. If the bidder logs onto the system in the middle of the bidding for a specific item, this area will not contain data until the NEXT ITEM is selected.
3. The base frame (containing bid buttons and the activity log) is updated each time an action is taken by the Clerk System, this bidder, or any other bidder during the bidding sequence for a particular item.

The bidder display also contains two "RELOAD" links on the display at all times. These links allow the individual bidder the ability to reload a particular area of the display:

RELOAD under the stream data – reconnects to the website URL that is broadcasting the audio/video stream.

RELOAD under the bidding frame – disconnects the bidder from the system and returns to the logon display for a bidder.

Figure 8A illustrates an example of a Bidders Display from the Cherokee Bid Engine. Figure 8B identifies the configuration required for the Bidder Device and Figure 9 defines the base process flow for the bidder function.

Table 4

MINIMUM BIDDER SYSTEM CONFIGURATION

IBM Compatible - Pentium 133MHz
16 MB RAM (32 MB RAM preferred)
CD ROM
Hard disk drive
1.44 MB Floppy drive
3COM 10/100 Ethernet NIC for high speed LAN connection
or 56Kbps modem with active phone line
Standard keyboard and mouse
Color display (800 x 600 24-bit color compatible)
Sound-Blaster compatible sound card with speakers*
*Compaq sound card is not compatible)

Windows 95, 98, or NT operating system
Netscape 4.05 or later
Netscape plug-in required by AMS software

Figure 9A illustrates Bidder Display after a URL connection is made. Figure 9B illustrates a Bidder Display with "user name" and "password" entered. Figure 9C illustrates a Bidder Display with a DISMISS message box. The bidder clicks on DISMISS to delete message box and to generate base Bidder Display shown in

Figure 6D. Figure 9D illustrates a Bidder Display after the login sequence is completed.

7 CLERK SYSTEM

The Clerk System is used to control the bid activities from both the floor and remote bidders in conjunction with the controlling Bid System. Entries made via the Clerk System result in display changes for all logged on bidders and the Marquee System. The action resulting from a particular entry is dependent on the "bid engine" being utilized by an individual auction. These engines are:

Cherokee

Iroquois

Mohawk

Apache (asking price model)

Figure 10A illustrates an example of a Clerk Display from the Cherokee Bid Engine. The configuration required for the Clerk System is shown in Figure 10B and in Table 5, and the base process flow for the system is contained in Figure 11.

Table 5

MINIMUM CLERK SYSTEM CONFIGURATION

IBM Compatible - Pentium 350MHz

32 MB RAM

CD ROM

4.0 GB Hard disk drive

1.44 MB Floppy drive
3COM 10/100 Ethernet NIC
Standard keyboard and mouse
VGA Display controller (1024 x 768)
Sound-Blaster compatible sound card with speakers*
(*Compaq sound card is not compatible)
17 inch CRT display (0.27 mm dot pitch)
Uninterruptible Power Source/Converter

Windows 95, 98, or NT operating system
Netscape 4.05 or later

Figures 11A/11B illustrate the initial Clerk System logon displays. Figure 11C illustrates the first DISMISS message box; the operator clicks on DISMISS to delete the message box. Figure 11D illustrates the second DISMISS message box; the operator clicks on DISMISS to delete message box. Figure 11E illustrates a Clerk System display after the login process is complete.

8 BID ENGINE PROCESSES

The bidding process involves the Clerk System, the Marquee System, and all bidders logged onto the auction. The specific process utilized for an individual auction is based on the "Bid Engine" selected by the auction to control this process. The base operation of the Bid Engines is identical in how the data packets are sent to the systems and the content of the data. Differences are primarily in the area of how remote bids are requested/accepted (either automatically or via a clerk entry), the options on the value to bid presented to

the Bidder Devices (one bid value = next higher sequential value versus five bid value options), and the ability to "set policy" changing bid increments based on the value of the last bid (increment always by a fixed value versus increment by $25 up to $3000, $50 up to $4000, and then by $100 increments).

The data sent to the Marquee System is basically the same for each of the Bid Engines. Logging displays are performed in the same manner in all Bid Engines, the content of specific messages varies by bid engine. Display format differences are in the bidding frame on the bidder device display and the base clerk display.

Each of the Bid Engine processes is defined in the following sections. Once the Marquee, Clerk, and Bidder Devices are active (logged on) and the base displays broadcast, the process becomes event driven based on a clerk or bidder entry on their respective systems. The process flows identify the actions taken by the Bid System in response to the initiation of these events.

8.1 CHEROKEE BID ENGINE

The following processes for the Cherokee Bid Engine are detailed in the process flow charts below. Once the Marquee, Clerk, and Bidder Devices have been initialized, the bid engine reacts to entries made by either the clerk or the bidder.

Initiation of first item or next item (Clerk function)

Enter starting bid (Clerk function) including +/- $500 button use

Enter floor bid (Clerk function)

Enter/accept remote bid (Bidder/Clerk function)

Sold Bid (Clerk function)

Request purchase info (Bidder function)

Delete Bid (Clerk function)

Message (Clerk function)

8.1.1 ACTIVATION OF THE BIDDING PROCESS FOR FIRST, NEXT, OR ANY ITEM

Figure 12 illustrates the activation of the Bidding Process for an Item. Figures 13A/B/C are examples of the Marquee, Clerk, and Bidder Device displays following the initiation of the first Item in the sequence. The clerk establishes the sequence of items based on the use of the NEXT ITEM bar on the display:

The first item can be selected by entering 1 and then clicking on NEXT ITEM; or clicking on NEXT ITEM if no other prior activity has occurred since the system was activated (the system presets to item 1 of the ringman subset data created from the pre-sales catalog in the **update_db.sh** process).

The next item is normally selected by clicking on NEXT ITEM. The next sequential entry defined in the ringman subset is selected.

Any item can be taken out of sequence by entering the ringman subset number and then clicking on NEXT ITEM. Successive NEXT ITEM entries are then based on the last item selected.

The NEXT ITEM entry is the single point that the "lower left" data is output to the bidder display and the four display areas on the Marquee System are updated for

the next item in the bidding process. These areas are not changed until NEXT ITEM is entered.

The sequence of items is determined from the pre-sales catalog information during the **update_db.sh** process. This process (defined in section 4.2) creates a "ringman subset" on the Bid System prior to the start of the auction. This data subset contains the run number, lower left data (specific fields from the pre-sales catalog data defined by the auction) and the starting bid if provided. During the NEXT ITEM process, the Bid System extracts the data from the ringman subset, based on the run number, and sends the necessary data to pre defined screen locations for the Marquee, Clerk, and Bidder Devices.

Figures 13A/13B/13C illustrate Marquee/Clerk/Bidder Displays after the first item is selected.

8.1.2 ENTER STARTING BID

Figure 14 schematically illustrates the entry of a starting bid. The clerk initiates the bidding process for each item by entering a starting bid for that item. This can be accomplished by two methods:

Enter a user name plus a value (numeric, no $ sign) in the two areas above the CHANGE BID bar. Then click on CHANGE BID. This will initiate the bidding by setting a starting value plus an entry in the activity log and setting the bid increment bars on both the clerk and the bidder

displays. The bid increment bars are preset for $100 increments for this bid engine.

Click on the +/- $500 bars until the desired starting value appears on one of the bid increment bars on the right side of the Clerk Display. Then click on the starting value bar. The system then generates the Clerk and Bidder Displays as done for the CHANGE BID entry.

From this point forward until the completion of bidding for a particular item (SOLD or NEXT ITEM), the CHANGE BID function can be utilized to enter a bid from a particular user or to jump the bid more than the amount shown on the last bid increment bar on the Clerk Display. If a user ID is not entered, the system defaults to "floor bidder." The CHANGE BID process can also be utilized to override a remote bid with an equivalent floor bid as clicking on the bid value automatically assigns the bid to an remote user if a bid has been made at the that price.

The +/- $500 bars are only active for the first bid on a specific item. After the initial (starting) bid is entered, these bars are inactive until SOLD or NEXT ITEM is entered on the Clerk Display.

Figure 13D/13E/13F illustrates the Marquee/Clerk/Bidder Displays after the starting bid.

8.1.3 ENTER FLOOR BID

Figure 15 schematically illustrates the entry of a floor bid. To enter a floor bid, the clerk follows one of two sequences:

- Click on the value of the floor bid on the increment bars if there is no bid from a remote bidder (flashing Marquee Display).
- Enter the value in the space above CHANGE BID and then click on CHANGE BID.

In either case the Marquee, Clerk and Bidder Displays are updated to reflect the accepted bid value and the increment bars reset for the next bid to be entered. The CHANGE BID sequence is typically used if the value of the bid to be accepted exceeds the highest value on the bid increment bars or the clerk enters a bid prior to a value being recognized by the auctioneer and the clerk must reset the bid to a lower value.

8.1.4 ENTER/ACCEPT/OVERRIDE REMOTE BID

8.1.4.1 ENTER REMOTE BID

Figure 16 schematically illustrates the entry of a remote bid. To enter a bid from a remote bidder, the bidder clicks the large value button that shows the next incremental bid value. The system validates the credit limit by adding dollars spent to this bid. If the value exceeds the pre-defined credit limit, a FUND message box is displayed on the Bidder Screen. The bidder then clicks on the

DISMISS area of the message box and the system resets the display such that a new bid could be entered. This FUNDS message box is regenerated each time the bidder enters a bid that would exceed the allowed total. This message is also displayed if a spectator (credit limit = $0) attempts to make a bid. If the bid is within the credit limit as defined above, the system sets the Marquee to flash and beep signifying an open remote bid has been entered. The activity logs are also updated on the clerk and bidder screens to reflect this condition.

If two or more remote bidders enter the same bid at the same time, the system takes the first bid received. Any other remote bidders have the display reset with the OUTBID message.

Figures 13G/13H illustrate Marquee and Clerk Displays when there is a Pending Remote Bid.

8.1.4.2 ACCEPT REMOTE BID

Figure 17 schematically illustrates the acceptance of a remote bid. To accept the remote bid, the clerk clicks on the increment bar containing that value (top bar on the right). An alternative method is to enter the remote user ID and the value and then click on CHANGE BID. For either sequence, the Marquee, Clerk, and Bidder Devices are reset to identify the accepted bid. This will stop the Marquee Display flashing/beeping.

Figure 13I is an illustration of an accepted remote bid on the Bidder Screen.

8.1.4.3 OVERRIDE REMOTE BID

Figure 18 schematically illustrates the override of a remote bid. If the same bid is received from both the floor and the remote bidder, the clerk can accept the remote bid as defined above or accept the floor bid by entering the value and clicking on CHANGE BID (the system defaults to "floor bidder" if no bidder ID is entered for the CHANGE BID function). The Bidder Display contains an OUTBID message if the floor bid was accepted. This will stop the Marquee Display flashing/beeping.

8.1.5 SOLD BID

Figure 19 schematically illustrates the process for a sold bid. To sell an item to either a floor or remote bidder, the clerk must first accept the final bid as discussed in sections 8.1.3 or 8.1.4. The clerk then clicks on the SOLD button to complete the sale for that item. The activity log and the Marquee Display identify the value and the remote bidder name/"floor bidder" as the person buyer of that item.

If the item was sold to a remote bidder, the system updates the user tables as follows:

The user file SPENT column is updated to reflect the sum of any prior sales to this user plus this sale such that subsequent credit limit checks are based on the current dollars spent by this remote user.

The table is updated to reflect the information for the item just purchased by the remote bidder (listing of each item purchased, value, and time). This data is then available to the remote user at any time during the auction by clicking on PURCHASE INFO at the top of the bidder display (section 8.1.6).

Figures 13J/K illustrate the Bidder/Clerk Screens after SOLD.

8.1.6 PURCHASE INFORMATION (REMOTE BIDDER)

Figure 20 schematically illustrates the remote bidder's process for requesting purchase information in the Cherokee Bid Engine. When a remote bidder clicks on the PURCHASE INFO button on the Bidder Display, the system links that bidder only to a new website page that contains the purchase history for that remote bidder for this auction. The format of the website page is shown below. To return to the Bidder Display, the user clicks on BACK or the X in the upper right corner of the website display.

username	656	
password	abcdefg	
purchase total	18400	
catalog number	**amount**	**time**
15	6500	09:26
27	11900	10:12

8.1.7 DELETE BID

Figure 21 schematically illustrates the process for deleting bids. The clerk is able to "delete" a bid by clicking on DELETE BID on the Clerk Display. The system deletes all data for that bid from the activity logs on the Clerk and Bidder Displays, resets the Marquee to the last accepted bid data, and resets the bid increment bars on the Clerk and Bidder Displays based on the last accepted bid prior to the bid being deleted.

8.1.8 MESSAGE FUNCTION

The message function is used by the clerk to send a message to either a single remote bidder or broadcast to all remote bidders. The message format defines the type of message to be processed. Figure 22A illustrates an example of a message screen. Figure 22B schematically illustrates the message process.

8.2 MOHAWK BID ENGINE

The following processes for the Mohawk Bid Engine are detailed in the process flow charts listed below. Once the Marquee, Clerk, and Bidder Devices have been initialized, the Bid Engine reacts to entries made by either the Clerk or the remote bidder.

Initiation of first item or next item (Clerk function)**

Enter starting bid (Clerk function) including +/- $500 button use**

Enter starting bid from Remote Bidder

Enter floor bid (Clerk function)**

Enter/accept remote bid (Bid/Clerk function)**

Sold Bid (Clerk function)**

Request purchase info (Bidder function)**

Delete Bid (Clerk function)**

Message (Clerk function)**

** functions same as Cherokee Bid Engine

The Mohawk Bid Engine is the same as the Cherokee Bid Engine described in the previous section with added capabilities for:

Entering a starting bid from a remote bidder

Four additional bid value buttons

Figures 23A/23B illustrate exemplary Bidder Screens and Figure 23C an exemplary Clerk Screen from the Mohawk Bid Engine.

8.2.1 MULTIPLE BID BUTTONS ON BIDDER SCREEN

The Mohawk Bid Engine has five bid bars on the Bidder Display; the main bar is for the next higher $100 increment from the last accepted bid while the remaining four bars are for values $200/$300/$400/$500 higher than the last accepted bid. The remote bidder clicks on the value desired to enter a bid in the same manner as previously described for the Cherokee Bid Engine. The Bid System recognizes which bar has been selected and updates the Marquee, Clerk, and Bidder Displays accordingly. Acceptance of the bid is handled the same as the

Cherokee Bid Engine. For this process, the Bid System must output five values to update the Bidder Display instead of a single value.

8.2.2 ENTRY OF A STARTING BID FROM THE BIDDER SYSTEM

To enter a starting bid from the Bidder Device, the bidder enters a value (all numeric, no dollar sign) and clicks on STARTING BID. The Clerk System then (a) accepts the starting bid by entering the value plus CHANGE BID or (b) overrides the remote starting bid by entering a different value + CHANGE BID. Once the bidding has been initiated for a specific item, the bidder starting value entry bar is deactivated until NEXT ITEM is selected on the Clerk System. The process is the same as that used to Enter, Accept, or Override a remote bid as previously defined for the Cherokee Bid Engine.

8.3 IROQUOIS BID ENGINE

The following processes for the Iroquois Bid Engine are detailed in the process flow charts listed below. Once the Marquee, Clerk, and Bidder Systems have been initialized, the bid engine reacts to entries made by either the clerk or the remote bidder.

Initiation of first item or next item (Clerk function)**

Enter starting bid (Clerk function) including +/- $500 button use**

Set Policy (Clerk function)

Enter starting bid from Remote Bidder**

Enter floor bid (Clerk function)**

Enter/accept remote bid (Bid/Clerk function)

Sold Bid (Clerk function)**

Request purchase info (Bidder function)**

Delete Bid (Clerk function)**

Message (Clerk function)**

** functions same as Cherokee or Mohawk Bid Engine

The Iroquois Bid Engine provides the following additional capabilities in addition to those of the Mohawk Bid Engine:

A specific button (REMOTE BID) must be selected to accept a remote bid, clicking on the bid value button accepts that same value from a floor bidder

SET POLICY – allows the auction to identify what bid increments are to be used based on the bid value (e.g., up to $2000 use $25 increments, for the next $2000 use $50 increments, above that level use $100 increments). This affects the process used by the Bid System to redisplay the clerk and bidder bar values as each bid is accepted.

Figures 24A/24B are example of a Bidder Display and a Clerk Display from the Iroquois Bid Engine.

8.3.1 ACCEPTANCE OF A REMOTE BID

Acceptance of a remote bid is accomplished by clicking on the REMOTE BID bar at the top of the Clerk Display. The process is the same as defined for the Cherokee Bid Engine for an accepted remote bid. If the Clerk clicks on the value

bar equivalent to the remote bid, the bid is assigned to the "floor". For this condition, the process is the same as the override process described for the Cherokee Bid Engine.

8.3.2 SET POLICY FUNCTION

The SET POLICY function is used to define the bid increments to be used on the bid increment bars on the Clerk and Bidder Displays for each item to be auctioned. The Clerk System cannot set starting bids or accept bids until the SET POLICY function has been completed.

The clerk clicks on the SET POLICY bar at the bottom left of the Display. This activates the set policy box on the Display. The Clerk then sets as many increment definitions as necessary by entering values in the two lines initially displayed plus ADDing as many line as necessary to complete the definition. Once all increments are defined, the Clerk clicks on SET POLICY in the message box to activate these values/increments in the bid processing. The server compares the value of the last accepted bid against these table values and generates the appropriate INCREMENT BID BARS for both the Clerk and Bidder Displays. This is done each time a bid is accepted from the floor or a remote bidder. The process of updating displays is the same as the Cherokee Bid Engine with the calculations performed by the server prior to the values being sent to the Displays.

8.4 APACHE BID ENGINE

The fourth Bid Engine utilized is Apache (or OGAC). In this Bid Engine, bids are "ASKed" for and then accepted by the clerk. A number of the processes are equivalent to those previously described with minor variations. The format of the Marquee and Bidder displays is different from those referenced in prior sections.

Figures 25A/25B/25C are sample Clerk, Marquee, and Bidder Displays from the Apache Bid Engine.

8.5 MARQUEE DISPLAY UPDATES

The Marquee is updated for the following actions:

The NOW SELLING LOT NUMBER field is updated when the clerk enters NEXT LOT. The field remains the same until the NEXT LOT button is selected.

The purple bar flashes (e.g., purple/green) when a remote bidder has made a bid. The flashing continues until the clerk accepts the bid value from either a floor bidder or a remote bidder.

REMOTE BID and REMOTE BID AMOUNT are updated when a remote bid has been made. These fields are cleared when the bid value is accepted (either floor or remote).

HIGH BIDDER and LAST LOT SOLD are updated each time the clerk clicks on the SOLD button. The HIGH BIDDER contains either FLOOR or the remote user ID.

The processes to update the Marquee are the same as those defined for the previous Bid Engines. The difference is the specific data elements and placement on the Display that vary between bid engines.

8.6 CLERK DISPLAY UPDATES

The clerk function is functionally the same as that defined for previous Bid Engines. The primary difference is that the clerk ASKs for a bid by clicking on one of the BID INCREMENT BARs on the right side of the screen. A bid for that value is then accepted by the clerk by clicking on either the FLOOR BID or REMOTE BID button at the top of the display. Actions taken/display updates for the clerk function occur as follows:

LOT NUMBER – updated when NEXT LOT is clicked for the next sequential entry in the ringman subset.

BIDDER – updated when a bid is accepted by the clerk.

SOLD AT – updated when the clerk selects the SOLD button.

ACTIVITY MESSAGEs – updated each time an action is taken by either the clerk or bidder.

CHANGE BID – allows the clerk to enter a value not shown on the BID INCREMENT BARs and to enter a starting bid.

DELETE BID – deletes the last bid and resets to the bid prior to the last bid.

NEXT LOT – same as the NEXT ITEM function.

REMOTE BID – clerk selects REMOTE BID to accept a bid from a remote bidder.

FLOOR BID – clerk selects FLOOR BID to accept a bid from the live gallery.

SOLD – clerk selects SOLD when after the last bid has been accepted and the auctioneer identifies the item as sold to that bidder. When the clerk clicks the SOLD button, a window is displayed on the Clerk Screen for the clerk to enter the bidder number to whom the item was sold. This data is then placed in the activity messages, the Marquee, and the bottom of the Bidder Screen.

The processes for these functions are the same as the Iroquois Bid Engine with the following differences:

The FLOOR BID button to accept a bid is used instead of clicking on the BID INCREMENT BAR for that value.

The BID INCREMENT BARs are used to broadcast a request for a bid rather than to accept a floor bid.

Figure 26 illustrates an example of a Clerk Screen from the Apache Bid Engine.

8.7 BIDDER DISPLAY UPDATES

Updates to the Bidder Screen and functions activated by the bidder include:

LOT # - updated each time NEXT LOT is selected by the clerk.

BIDDER/CURRENT BID – updated each time the clerk accepts either a FLOOR or REMOTE BID.

BID HISTORY – updated each time the bidder or clerk takes an action.

BID STATUS – indicates status of the remote bid from this user;

YELLOW = PENDING, bid has been selected by the bidder

GREEN = ACCEPTED, bid was accepted by the clerk

RED = OUTBID, the FLOOR or another remote bid was accepted for this value.

PREVIOUS LOT #/WINNING BIDDER/WINNING BID AMOUNT – updated each time the clerk clicks the SOLD button.

BID VALUE – updated each time the clerk ASKs for a bid. The bidder clicks on this button to enter a bid. The button is cleared each time the SOLD button is selected by the clerk and remains clear until the NEXT LOT first ASKing price is entered by the clerk.

HELP – activates a new webpage with hints/FAQs for the bidder. Bidder returns to the normal bid display by clicking the X in the upper right corner of that webpage.

Figure 26 illustrates an example of an updated Bidder Screen.

All ringman subset data for the bidder display is generated at one time for the display and placed to the right of the Bidders Display. This area is scrollable and

contains limited information for each lot being sold. The data is broadcast to the bidder when the logon sequence is completed.

9 DATA MINING

Upon completion of the auction, the bidders log and the bid log are available for post processing from the Bid System directly or through a predefined URL. The data is collected during the auction by the Bid System as the events take place.

The bidders log must be downloaded directly from the Bid System.

The bid log is obtained by accessing the required URL for that auction. A script is executed that extracts the data from the Bid System and places it on the assigned website.

9.1 SCRIPT PROCESS – BIDDERS LOG

The Bid System maintains a log of bidders who connect to each auction. The log entries are maintained in a table on the System with an entry being made each time a bidder, the clerk, or the Marquee are initiated through the login process. The **showlogins** script extracts the data from this table and places the data on screen. This script requires a unique logon/password combination to be entered on the Bid System to be initiated. The analysis of this data identifies the user logon ID and the period for which the user was connected to the System.

9.2 SCRIPT PROCESS – BID LOG

The access to the bid log is via a special website specifically set up to extract the data from the bid server. The URL for this process is

http://onlineringman.com/cgi-bin/ringman/query.cgi?bidlog.html+dbname-auction

where "auction" is a unique name for each auction.

When the website is accessed, the script initiated from the website performs a database query to the Bid System. The Bid System extracts the log entries from internal tables and places them in sequential time order on the website display. Automated analysis provides remote bidder information including bids made per user ID, number of items purchased by user ID, time of sale, amount of sale, average $value of items sold to remote versus floor bidders, and total statistics for the auction. The analysis can be defined specifically for each auction from the unique data items available.

10 AUDIO/VIDEO SYSTEM

The audio/video system of the present invention ("A/V System") is a Client/Server audio and video transport system. It provides a streaming audio and video feed from a customer site to client workstations that may be located anywhere on the global Internet. Clients receive the A/V feeds using standard WWW browser software and sound cards. The system may be used for standalone applications, or may be incorporated into interactive web-enabled database applications as a functional subset.

There are two overarching design elements that firmly define and delineate the unique nature of the A/V System. **Connectionless, Non-Buffered Performance.** Mass-market audio/video streaming applications are oriented

towards delivery of content under the assumption that buffered delay is acceptable in order to ensure very high quality (CD-quality audio, picture-quality video) at the client end of the application. While appropriate for consumer and high-end business delivery, this approach misses a potentially large audience that will trade a degree of final output quality for a more real-time performance experience (for example, to receive the "live" look and feel of an auction). These mass-market solutions are typically connection-oriented in their network delivery techniques.

The A/V System uses connectionless, non-buffered designs that, in spite of the implications of the terminology, delivery FM-quality (voice) audio and still video streams to remote clients while at the same time ensuring an interactive response turnaround from the client application of one second. A bonus of this approach is that it functions quite well using the lowest common denominator of Internet access transport at this time -- the V.90 analog modem connection. As end-user connection technology migrates to schemes with speeds higher than 56 Kbps (asynchronous) -- for instance, to ADSL or cable modem technology -- the performance inherent in the A/V System design will already be guaranteed.

10.1 INNOVATIVE APPLICATION OF CURRENT A/V TECHNOLOGIES

The A/V System provides its audio, video, and integration services by using existing transport and display services in a completely unique manner. A standard audio encoding/decoding scheme is extended to ensure the reliable transmission of an FM-quality stream. A still video standard – JPEG – is

sampled and streamed to simulate a live video feed while eliminating the overhead and bandwidth requirements that typically accompany such a configuration. And the final client-side presentation is integrated into standard WWW browser technology, removing the need for custom standalone applications on the remote end of the service.

10.2 A/V SYSTEM OVERVIEW

When considering the development of a system that can deliver an audio and video feed from a source server location to multiple client locations, there is one critical architectural decision that must be investigated, that of selecting the Core Network Topology. This decision not only drives the final system architecture, it also determines the audience and market scope that may be addressed by the final system.

The Core Network Topology is, in short, the choice of a transport backbone over which captured audio and video streams will be processed and delivered to remote client applications. There are many choices for such a backbone, including:

Private Broadcast/Multicast [Terrestrial or Satellite] Networks

Proprietary [Branded] Internet Broadcast Channels and Software

Custom Standards-Based Internet Software

Of these topology choices, the first, like broadcast or cable television, would result in a system suited to content delivery on a large scale, but from a practical

standpoint unaffordable by small to medium businesses that wished to deliver content. The second, while less restrictive than the first, typically carries licensing costs, co-branding requirements, and is optimized for one-way content transport. Because the A/V System needs to interact with other interactive database and web solutions, custom development of the A/V System using Open Source, Open Systems Standards was logically mandated.

The System is designed under the assumption that the majority of remote clients being served by an installation are connecting to the Internet using V.90 (56K) modems. It is further assumed that, on average, a V.90 connection will receive roughly 40 Kbps of reliable bandwidth over time during a session. Using these assumptions, general design parameters may be defined for the audio and video streams that will be sent to client software by the server:

10.1.1 AUDIO

The audio stream must be structured such that it can maintain a voice-quality feed under conditions that may include limited bandwidth (i.e. less than 56 Kbps sustained). In order to minimize development and, hence, customer costs, the audio encoder and server must run in an Open Systems server environment, and the client application should require minimal development using off-the-shelf browser technology.

The selected audio encoder/decoder utilizes the GSM 06.10 codec library. The streaming audio produced by this library utilizes 13.3 Kbps of bandwidth and delivers an 8 KHz voice-quality audio signal to the remote bidder client machine's sound card.

10.1.2 VIDEO

Requirements for the video stream are the same as those outlined above for audio, with one subtle difference. While the long-range architecture of A/V System should not, and does not, preclude the inclusion of non-lossy or full-motion video technology; initial design requirements are such that video need not be completely full-motion. A rate of one frame per three seconds has been set as a reasonable benchmark for the current release of the A/V System.

The video encoder utilizes JPEG for video software, and the streamer portion of the server is designed to give priority, should it be required, to the audio feed. The combination of the audio and video stream to any given remote bidder client will, under normal network conditions, fit into the 40 Kbps average ceiling estimated for V.90 connections.

The A/V System is composed of hardware and software elements configured in a client/server architecture. The server side of the A/V System is configured to run in the Linux Operating System environment, and consists of an Encoding Server (deployed at the site of audio/video capture) and a Master Server (maintained at a secure high-capacity AMS Point-of-Presence). The Encoding Server sends a

single audio and video stream to the Master Server, which in turn oversees distribution of multiple streams to the remote clients. The client side of the A/V System is configured to run with WWW/HTML browsers; currently the client is deployed specifically for the Netscape™ 4.X browser release, but the architecture does not preclude compiling for Microsoft Internet Explorer™ or other browsers.

Figures 28 and 29 present general views of the layout and operation of the A/V System.

10.3 ENCODING SERVER – FUNCTIONAL

The A/V System Encoding Server runs under Linux and uses the Osprey 100 video capture card (with the BTTV driver for Linux) for video capture. A Linux-compatible 16-bit sound card (e.g. SB16, SB32, ESS, etc) is required for audio capture. The encoder opens up two connections to the Master Server – one for a video stream and one for an audio stream. The video is compressed into a full JPEG image and is sent at a steady rate, determined at server connection. The audio is compressed using GSM 06.10 and is sent at a steady rate, also determined at server connection.

The encoder uses a configuration file "ams-streamer.conf" which is in the format of:

ID server-port audio-interface video-interface server-address password

Only the first line of the file will be read; all other lines are ignored.

Valid configurations include:

123 9781 - - 192.168.1.1 password

456 9782 ppp0 ppp1 192.168.1.1 pw

Note that the IP address used for the server-address field above is a dummy example only.

The *123* configuration uses "-" (meaning default interface) for both audio and video interfaces; the *456* configuration uses ppp0 for audio and ppp1 for video. For dual-modem configurations (possible with multilink-PPP configurations, but not recommended by AMS), it is desirable to select one modem to send audio and the other to send video. Doing so will require proper setup of the routing tables and root privileges (to allow selection of which interface to send the packets through), and can be done by making the encoder program setuid-root.

The encoder is invoked by running *streamer* from the Linux command line. [Note – In A/V Transport versions to date there are two other programs – *streamer-save* and *streamer-play* – both of which are currently for testing purposes only.] The *streamer* program accepts/has the following command-line options:

streamer –f frequency –q quality -h

The **–f frequency** option specifies the frequency of video capture in tenths of a second, with the default set at 30. The **–q quality** option specifies the JPEG compression quality as a relative percentage between 1 and 100, where lower numbers equate to lower image quality, with the default set at 15. The **–h** option will display command-line help.

10.4 ENCODING SERVER – BUILD ENVIRONMENT

The encoder runs under Linux and requires GNU make and GNU gcc. It uses the Independent JPEG Group's (IJG) JPEG software (for video), the GSM 06.10 library by Jutta Degener and Carsten Bormann of Technische Universitaet Berlin (for audio), and the MD5 message digest algorithm by RSA Data Securities, Inc. (for authentication).

The Build Environment specifics are:

Operating System:	Red Hat Linux 5.2 (or higher)
Linux Kernel:	2.2.X (or higher)
Kernel Library:	libc.so.6 (or higher)
Compiler:	GNU C 2.7.2 (or higher)

10.5 ENCODING SERVER – HARDWARE ENVIRONMENT

The Encoding Server requires the following hardware components:

IBM compatible PC, Intel Pentium 450 MHz (or faster) CPU

256 MB RAM (minimum)

1.0 GB Hard Disk (minimum)

V.90 (56K) internal modem

10/100 Ethernet Network Interface Card

Keyboard and Mouse

Osprey 100 Video Capture Card

15" CRT Display (.27 mm dot pitch)

Sound card (SB16 compatible) with speakers and microphone input

CD ROM drive

3.5" 1.44 MB floppy drive

Video Capture Device (Sensormatic camera preferred)

Audio Line Mixer (Radio Shack recommended)

A routed Internet connection using a high-capacity circuit technology is suggested. (Single-link analog PPP connectivity via the internal modem is possible, but not supported by AMS. Multilink analog PPP connectivity via dual internal modems is also possible, but again is not actively supported.) The Internet connection may use any of several available connection technologies, including but not limited to (a) dialup ISDN (128K dual-channel), (b) dedicated ISDN (128K), (c) xDSL, (d) Frame Relay, or (e) dedicated private line.

10.6 MASTER SERVER – FUNCTIONAL

The Master Server accepts streams from Encoding Server and sends them to clients. It supports multiple streams. Each stream is protected by a password to prevent unauthorized encoders from using the server. With the current implementation, a stream is ready to send to clients when both the audio and video channels have been established. Ideally (perfect network conditions), packets are sent using the audio packets as the pulse (~200ms) and follow by a set video frame size. Audio and video are delivered to clients in a single channel to ensure more priority for the audio and to lessen the amount of buffering required for the audio. Audio is not considered critical (lost packets allowed) on both encoding and client end. This is by design as the entire A/V System is constructed for integration with interactive database applications (live auctions are a primary example). Under network congestion or error conditions the integrated application must receive higher network priority.

Video on the encoding end is not considered critical. However, video on the client end is considered critical (any missing parts will be re-requested until all parts are available; note that images that seem to be incomplete are due to the encoder, not the client). Remote bidder clients will receive the most recent complete frame available on the server at the time of request. Currently, connections to streams do not have access controls to limit which connection can view which stream.

The server uses a configuration file **ams-server.conf**, which is in the format of:

ID server-port stream-port password

Multiple entries may be specified to allow multiple streams (the maximum number of streams allowed is set up at program compile time). The ID, server-port, and stream-port must be unique. [Note: the current implementation uses port 9780 for all server ports, regardless of the number specified.]

The server is invoked by running **server** at the Linux command line. The **server** process should be left running while the Master Server machine is active. At this time the server program has not been configured to run as a daemon process under the control of inetd. The server program should be run by a non-root user. It is suggested that the **screen** program be used to start the server:

$ screen ./server

This will allow the program process to be detached from the console or terminal session using the key sequence **Ctrl-A + D**. The session may be reattached at a later time with the command:

$ screen –r {pid}

Where {pid} is the process ID of the server program.

If the server code was compiled with "TESTING" defined, the command-line "-t clients" is available for stress testing with "clients" amount of fake clients.

10.6 MASTER SERVER – BUILD ENVIRONMENT

The Master Server requires GNU make and GNU gcc. It uses the MD5 message digest algorithm by RSA Data Securities, Inc. for authentication.

The Build Environment specifics are:

Operating System:	Red Hat Linux 5.2 (or higher)
Linux Kernel:	2.2.X (or higher)
Kernel Library:	libc.so.6 (or higher)
Compiler:	GNU C 2.7.2 (or higher)

10.7 MASTER SERVER – HARDWARE ENVIRONMENT

The Master Server requires the following hardware components:

IBM compatible PC, "Server" Class

Dual Intel Pentium-II 450 MHz (or faster) CPUs

256 MB RAM (minimum)

4.0 GB Hard Disk (minimum)

100 Mbps Fast Ethernet Network Interface Card

Keyboard and Mouse

15" CRT Display (.27 mm dot pitch)

CD ROM drive

3.5" 1.44 MB floppy drive

10.8 CLIENT – FUNCTIONAL

The client is a Netscape plug-in using Win32 code. It opens up a connection to the server and decodes the packets it receives into audio and video. The audio is played through the default Windows audio playback device. The video is displayed in the Netscape browser page. The client will try to maintain a constant three audio buffers for playback (~700ms). If it has fewer, it will play the audio a bit slower; if it has more, it will cut off the audio if there are too many, or it will play the audio a bit faster to catch up.

The client must start with "np" and end with ".dll" and reside in Netscape's plug-ins directory.

The client may be invoked from an HTML page by using the following:

<embed src=192.168.1.1:9873 width=256 height=192 type="application/x-mis-ams"></embed>

The src parameter should point to the server's IP address and the streamer's port.

10.9 CLIENT – BUILD ENVIRONMENT

The client runs under Netscape and Windows 9x/NT. It requires MS Visual C++ (load up the amsclient project and compile the NS project; the resulting npams.dll

file from the NS project is the plugin to use). It uses the Independent JPEG Group's (IJG) JPEG software (for video), the GSM 06.10 library by Jutta Degener and Carsten Bormann of Technische Universitaet Berlin (for audio), the MD5 message digest algorithm by RSA Data Securities, Inc. (for authentication), and Netscape's plug-in SDK.

The Build Environment specifics are:

Operating System: Windows 95/98/NT

Compiler:Microsoft Visual C++ - current release

10.10 COMPONENT LICENSE INFORMATION

GSM 06.10 LIBRARY:

Copyright 1992, 1993, 1994 by Jutta Degener and Carsten Bormann, Technische Universitaet Berlin

JPEG LIBRARY:

This software is copyright (C) 1991-1998, Thomas G. Lane.

MD5 LIBRARY:

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

It is believed that the following unique attributes distinguish the A/V System from other currently available Internet-based audio/video transmission systems.

10.11 ENGINEERING DESIGN

10.11.1 SPECIFIC VS. MASS MARKET DESIGN

Other A/V products are designed for very specific market applications, and the underlying architecture tends to be reflected in the final design, customer implementation costs, and (often) branding requirements. These design categories include One-to-One Transport, Group Collaboration Transport, and One-to-Many Transport. Products like IP-Phone packages and online meeting tools exemplify the first two categories. These products tend to be low cost (often due to branding/advertising requirements), but are not designed to scale well beyond point-to-point or small group usage. Most are, as well, audio-centric in their design. The third category is oriented towards broadcast or multicast to larger audience size, but products to date either tend of focus on delivery of pre-recorded content or, where live delivery is concerned, involve extremely high cost server configurations that keep the tools out of reach for small to medium sized businesses. (Co-branding is sometimes used as a way to alleviate costs, but many firms that would like to use audio/video transport solutions are reluctant to enter co-branding or advertising oriented agreements.)

The A/V System, while scalable for very large installations, was designed with the small to medium size market in mind. The price-performance profile for the System is, then, a unique point of differentiation in the marketplace.

10.11.2 CODEC AND TRANSPORT SELECTION

If an audio/video system is engineered to deliver audio at CD-quality levels, and video at good approximations of broadcast-quality (or at least moderate video-conferencing quality), it will be engineered with codec and transport mechanisms that are "highly reliable" and that tend to use some form of carefully designed buffering for incoming streams to the client application. The term "highly reliable," however, should not be taken to infer that a design using "less reliable" delivery mechanisms or "lossy" codec schemes is as a result a less robust design. The A/V System was initially designed to deliver the "look and feel" of a live auto auction to an online Internet-based audience, and the codec and transport design decisions reflect the unique requirements of this market.

In a live auto auction, there is an inherent degree of "noise" as a part of the process itself; product is rolled through the auction lane at a rate no slower than one vehicle every three minutes, and bidders have had time prior to the actual sale to investigate and inspect the vehicles. During the live sale process, a bidder is highly likely to focus on the progress of a particular bid by mentally parsing the increasing bid numbers from the rapid pace of the auctioneer's patter. Under such conditions the "audio and video stream quality" become secondary considerations. When moving such an experience to an Internet-based environment, the A/V System engineering design is allowed to reflect these same conditions, hence the fact that A/V System delivers audio and video streams, but not in such a manner that they would interfere with the ability of the

remote bidder client to see the bid progress (in this case, via a database update sent to the browser from a bid server) and act accordingly (by entering a remote bid, for instance).

The GSM 06.10 codec is "lossy." It will not reproduce CD-quality audio but, given the design and market requirements, does not need to for the A/V System to function properly. The quality and frequency settings of the JPEG encoder, and the transmission/retry mechanisms built into the Encoding Server's streamer, do not reproduce a broadcast-quality video signal but, once again, do not need to do so. In consideration of marketplace drivers and requirements for integration with larger system designs, the codec and transport design of the A/V System represents a unique solution.

10.11.3 STREAM PRIORITIZATION

A follow-on to the fundamental idea of codec and transport selection is the overall design of stream prioritization within the A/V System. Working with an estimated average bandwidth of 40 Kbps, the streamer software itself is written so that packets from the encoder's GSM stream will receive consistent priority and handling unless video is available for processing. On the client side, the browser applet is programmed to follow these same rules, except it is also configured to look for packets from a separate bid server (under the assumption that A/V Transport is installed as a turnkey live auction solution), and if it sees these packets to give them priority over both audio and video streams.

11 PLATFORM INTEGRATION

11.1 SOFTWARE & HARDWARE PLATFORM SELECTION

The A/V System is designed to support several unique market niches, and the software design and operation (as discussed above) reflect this philosophy. However, the System is also engineered to be flexible and extensible should custom implementation opportunities present themselves. In order to ensure that developers are able to respond to rapidly changing conditions both in terms of Internet topology or services and in terms of market demand, the bulk of the software platform upon which A/V System is constructed relies on highly available "Open Source" operating system environments and language toolkits.

Similarly, the selection of hardware components was focused on the IBM/Intel-compatible market on the assumption that the best overall cost efficiencies for small to medium sized environments is to be found with this type of equipment. The server software (encoder and main server) may be ported to more proprietary environments if needed (Sun hardware and the Solaris OS, for examples), but to date the more open platform has proven quite satisfactory in terms of handling customer load. The software and hardware platform flexibility, however, should be noted as contributing to the unique nature of the A/V System of the present invention.